

Title	ソフトウェア信頼度成長を考慮したコンピュータシステムの性能評価モデル (不確実・不確定性下での意思決定過程)
Author(s)	高橋, 淳一; 得能, 貢一; 山田, 茂
Citation	数理解析研究所講究録 (2010), 1682: 225-232
Issue Date	2010-04
URL	http://hdl.handle.net/2433/141380
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

ソフトウェア信頼度成長を考慮したコンピュータシステムの性能評価モデル

鳥取大学大学院・工学研究科 高橋 淳一 (Junichi Takahashi)[†]

鳥取大学大学院・工学研究科 得能 貢一 (Koichi Tokuno)[†]

鳥取大学大学院・工学研究科 山田 茂 (Shigeru Yamada)[†]

[†]Graduate School of Engineering, Tottori University

1 はじめに

近年、ハードウェアおよびソフトウェア相互の性能/品質を同時に考慮しながら、コンピュータシステムの最適設計を行うハードウェア/ソフトウェアコデザイン (hardware/software codesign, 以下、単にコデザインと略す [1], [2]) と呼ばれる設計思想に注目が集まっている。これに伴って、コンピュータシステムの信頼性/性能評価法においても、コデザインの概念の導入が試みられている。Goel and Soenjoto [3] や Tokuno and Yamada [4] は、ソフトウェアシステムの信頼度成長過程を考慮して、マルコフ過程に基づくハードウェアソフトウェアシステム (以下、単にシステムと略す) に対する信頼性/可用性評価モデルを構築した。Sumita and Masuda [5] は、システム信頼性/可用性評価尺度を近似的に算出するために、ラゲール変換 [6], [7] を適用した。一方、フォールトトレラント性を考慮したシステム信頼性/性能評価モデルもよく議論されている。Osaki et al. [8] は、2 ユニットハードウェアシステムに対してソフトウェア信頼度成長過程を考慮した上で、2 種類の復旧シナリオがシステムアベイラビリティ評価に与える影響を調査した。Welke et al. [9] は、3 重冗長ハードウェアシステムに対して、代表的な非同次ポアソン過程 (NHPP) モデルである Goel and Okumoto [10] モデルを適用することによって、ソフトウェアの信頼度成長過程を記述した。Lai et al. [11] は、分散システムに対するアベイラビリティ解析を行った。上記 3 つの研究は主にハードウェアシステムの冗長化に着目していたのに対して、Levitin [12] は冗長ソフトウェアシステム [13], [14] の可用性解析を行った。

本論文では、上記に紹介した既存研究とは異なるアプローチで、システムのパフォーマンス (performability) 評価モデルについて議論する。特に、「予め定められた時間内に仕事の処理を終了することができる性質や度合い」と定義されるリアルタイム性 (実時間性, real-time property) [15] を考慮する。システム自身の時間的振舞いは、Tokuno and Yamada [4] のモデルを用いて記述する。システムの形態はマルチタスクシステムを想定し、仕事の多様性も考慮する。すなわち、仕事の到着過程を NHPP で記述し、各仕事に課せられる処理時間制約は確率変数として取り扱う。処理時間制約以内に処理が完了する仕事数の分布は、無限サーバ待ち行列理論 (infinite-server queueing theory) [16], [17] を用いて解析される。

本論文の構成を以下に示す。第 2 節では、基本モデルとなる Tokuno and Yamada [4] のモデルを概説する。第 3 節では、システムの動作規定を定義して、処理が完了する仕事数の分布を解析する。第 4 節では、システムの性能評価尺度を、時間およびソフトウェアシステムに施されるデバッグ作業回数の関数として導出する。第 5 節では、本手法に基づくシステムの性能解析例を示す。最後に第 6 節では、本論文で得られた成果をまとめる。

2 コンピュータシステムの可用性評価モデル

本論文の議論の基礎となるシステムの可用性評価モデルを構築するための仮定を、以下に掲げる。

- (A1) システムにソフトウェア故障もしくはハードウェア故障が発生すると、システムダウンとなり、直ちに修復作業が開始される。また、修復作業が完了するまで、システムは動作することができない。
- (A2) ソフトウェアサブシステムの修復作業には、デバッグ作業が含まれる。このとき、完全デバッグ率 a でフォールトの修正に成功し、確率 $b (= 1 - a)$ で失敗する。デバッグ作業が成功したとき、1 つのフォールトがソフトウェアサブシステムから修正除去され、ソフトウェアサブシステムの信頼度が成長する。
- (A3) 修正されたフォールト数が n 個のとき、次のソフトウェア故障発生時間間隔 $U_n^S (n = 0, 1, 2, \dots)$ および修復時間 $V_n^S (n = 0, 1, 2, \dots)$ は、それぞれ平均 $1/\lambda_n$ および $1/\mu_n$ の指数分布に従う。 λ_n および μ_n は n の非増加関数とする。
- (A4) ハードウェアサブシステムに対する故障発生時間間隔 U^H および修復時間 V^H は、それぞれ平均 $1/\theta$ および $1/\eta$ の指数分布に従う。

時刻 t におけるシステムの状態を表す確率過程 $\{X(t), t \geq 0\}$ の状態空間を、以下のように定義する。

$W = \{W_n : n = 0, 1, 2, \dots\}$: システムは動作している状態,

$R^S = \{R_n^S : n = 0, 1, 2, \dots\}$: ソフトウェア故障が発生し、その修復作業のためシステムは動作できない状態,

$R^H = \{R_n^H : n = 0, 1, 2, \dots\}$: ハードウェア故障が発生し、その修復作業のためシステムは動作できない状態。

ここで、 $n(=0, 1, 2, \dots)$ は、各状態において修正されているフォールト数を表す。

仮定 A2 より、システムが $\{X(t) = R_n^S\}$ にあるときに修復作業が完了すると、

$$X(t) = \begin{cases} W_n & (\text{確率 } b : \text{不完全デバッグの場合}) \\ W_{n+1} & (\text{確率 } a : \text{完全デバッグの場合}) \end{cases}, \quad (1)$$

と遷移する。 $Q_{A,B}(\tau) (A, B \in \{W, R^S, R^H\})$ は、 $\{X(t), t \geq 0\}$ が時間区間 $(0, \tau]$ の間に状態 A から状態 B に遷移する 1 ステップ推移確率を表すものとする、 $Q_{A,B}(\tau)$ は以下のように与えられる。

$$Q_{W_n, R_n^S}(\tau) = \Pr\{U_n^S < U^H, U_n^S \leq \tau | X(0) = W_n\} = \frac{\lambda_n}{\theta + \lambda_n} [1 - e^{-(\theta + \lambda_n)\tau}], \quad (2)$$

$$Q_{W_n, R_n^H}(\tau) = \Pr\{U^H < U_n^S, U^H \leq \tau | X(0) = W_n\} = \frac{\theta}{\theta + \lambda_n} [1 - e^{-(\theta + \lambda_n)\tau}], \quad (3)$$

$$Q_{R_n^S, W_{n+1}}(\tau) = \Pr\{V_n^S \leq \tau, X(V_n^S) = W_{n+1} | X(0) = R_n^S\} = a(1 - e^{-\mu_n \tau}), \quad (4)$$

$$Q_{R_n^S, W_n}(\tau) = \Pr\{V_n^S \leq \tau, X(V_n^S) = W_n | X(0) = R_n^S\} = b(1 - e^{-\mu_n \tau}), \quad (5)$$

$$Q_{R_n^H, W_n}(\tau) = \Pr\{V^H \leq \tau | X(0) = R_n^H\} = 1 - e^{-\eta \tau}. \quad (6)$$

図 1 に、確率過程 $X(t)$ の状態遷移図を示す。

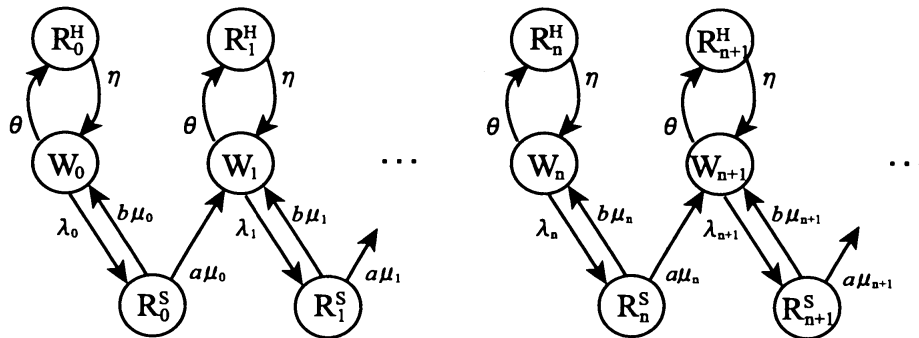


図 1 : $X(t)$ の状態遷移図

まず、 $X(t)$ の状態 W_i から状態 W_n への初到達時間間隔 $S_{i,n} (i \leq n)$ の分布関数 $G_{i,n}(t) \equiv \Pr\{S_{i,n} \leq t\}$ を導出する。このとき、 $G_{i,n}(t)$ に関する再生方程式は、

$$G_{i,n}(t) = Q_{W_i, R_i^S} * Q_{R_i^S, W_{i+1}} * G_{i+1,n}(t) + Q_{W_i, R_i^S} * Q_{R_i^S, W_i} * G_{i,n}(t) \\ + Q_{W_i, R_i^H} * Q_{R_i^H, W_i} * G_{i,n}(t) \quad (i = 0, 1, 2, \dots, n-1), \quad (7)$$

と得られる。ただし、 $*$ はラプラスたたみ込みを表す。式 (7) にラプラス-スチルチエス変換 (Laplace-Stieltjes transform, 以下 L-S 変換と呼ぶ) [18] を適用し、再帰的に解くと、 $G_{i,n}(t)$ は、

$$G_{i,n}(t) \equiv \Pr\{S_{i,n} \leq t\} \\ = 1 - \sum_{k=i}^{n-1} (A_{i,n}^1(k)e^{-d_k^1 t} + A_{i,n}^2(k)e^{-d_k^2 t} + A_{i,n}^3(k)e^{-d_k^3 t}) \quad (t \geq 0; i, n = 0, 1, 2, \dots; i \leq n), \quad (8)$$

と与えられる。ここで、 $-d_k^1$ 、 $-d_k^2$ および $-d_k^3$ は、次の s に関する 3 次方程式

$$s^3 + (\theta + \eta + \lambda_k + \mu_k)s^2 + (\theta\mu_k + \eta\lambda_k + \eta\mu_k + a\lambda_k\mu_k)s + a\eta\lambda_k\mu_k = 0, \quad (9)$$

の異なる 3 つの実数解を表す。また、係数 $A_{i,n}^1(k)$ 、 $A_{i,n}^2(k)$ および $A_{i,n}^3(k)$ は、それぞれ

$$A_{i,n}^1(k) = \frac{\prod_{j=i}^{n-1} a\lambda_j\mu_j(\eta - d_k^1)}{d_k^1 \prod_{\substack{j=i \\ j \neq k}}^{n-1} (d_j^1 - d_k^1) \prod_{j=i}^{n-1} (d_j^2 - d_k^1)(d_j^3 - d_k^1)} \quad (k = i, i+1, \dots, n-1), \quad (10)$$

$$A_{i,n}^2(k) = \frac{\prod_{j=i}^{n-1} a\lambda_j\mu_j(\eta - d_k^2)}{d_k^2 \prod_{\substack{j=i \\ j \neq k}}^{n-1} (d_j^2 - d_k^2) \prod_{j=i}^{n-1} (d_j^3 - d_k^2)(d_j^1 - d_k^2)} \quad (k = i, i+1, \dots, n-1), \quad (11)$$

$$A_{i,n}^3(k) = \frac{\prod_{j=i}^{n-1} a\lambda_j\mu_j(\eta - d_k^3)}{d_k^3 \prod_{\substack{j=i \\ j \neq k}}^{n-1} (d_j^3 - d_k^3) \prod_{j=i}^{n-1} (d_j^1 - d_k^3)(d_j^2 - d_k^3)} \quad (k = i, i+1, \dots, n-1), \quad (12)$$

と与えられる。ここで、 $\sum_{i=0}^{-1} \cdot$ および $\prod_{\substack{i=0 \\ j \neq i}}^0 \cdot = 1$ であるとする。このとき、

$$\sum_{k=i}^{n-1} [A_{i,n}^1(k) + A_{i,n}^2(k) + A_{i,n}^3(k)] = 1, \quad (13)$$

が成り立つことに注意する。式 (8) は、hypoexponential 分布 [17] である。付録 A に、 $G_{i,n}(t)$ の導出過程の詳細を示す。

$P_{W_i, W_n}(t) \equiv \Pr\{X(t) = W_n | X(0) = W_i\}$ ($i \leq n$) は、 $\{X(t), t \geq 0\}$ の状態 W_n に関する状態占有確率を表すものとする。このとき、 $P_{W_i, W_n}(t)$ に関する再生方程式は、

$$P_{W_i, W_n}(t) = G_{i,n} * P_{W_n, W_n}(t), \quad (14)$$

$$P_{W_n, W_n}(t) = e^{-(\theta + \lambda_n)t} + Q_{W_n, R_n^S} * Q_{R_n^S, W_n} * P_{W_n, W_n}(t) \\ + Q_{W_n, R_n^H} * Q_{R_n^H, W_n} * P_{W_n, W_n}(t), \quad (15)$$

と得られる。式 (14) および (15) を解くと、 $P_{W_i, W_n}(t)$ は、

$$P_{W_i, W_n}(t) \equiv \Pr\{X(t) = W_n | X(0) = W_i\} = \frac{g_{i,n+1}(t)}{a\lambda_n} + \frac{g'_{i,n+1}(t)}{a\lambda_n\mu_n}, \quad (16)$$

と得られる。ただし、 $g_{i,n}(t) \equiv dG_{i,n}(t)/dt$ 、 $g'_{i,n}(t) \equiv d^2G_{i,n}(t)/dt^2$ である。付録 B に、 $P_{W_i, W_n}(t)$ の導出過程の詳細を示す。

3 システムの動作規定および性能評価モデルの解析

システムに到着する仕事の確率的振舞いや仕事の処理時間に関する仮定を以下に示す。

- (B1) システムが同時に処理できる仕事数は、十分大きい。
- (B2) 時刻 t までにシステムに到着する仕事数 $\{N(t), t \geq 0\}$ は、到着率 $\omega(t)$ および平均値関数 $\Omega(t) \equiv E[N(t)] = \int_0^t \omega(x)dx$ の NHPP に従う。
- (B3) 各仕事の処理時間 Y は、独立で同一の一般分布 $F_Y(y) \equiv \Pr\{Y \leq y\}$ に従う。
- (B4) 各仕事には、処理時間制約 T_r が課せられており、一般分布 $F_{T_r}(t) \equiv \Pr\{T_r \leq t\}$ に従う。
- (B5) 各仕事の処理時間が処理時間制約 T_r を越えるとき、あるいは仕事の処理中にシステム故障が発生したとき、その仕事の処理はキャンセルされる。

図 2 に、システム故障時間、仕事の処理時間および処理時間制約と仕事処理の完了/破棄の関係を示す。ここで、図中の $U_n^{sys}(n = 0, 1, 2, \dots)$ は、フォールト修正数が n 個のとき、次のシステム故障発生時間間隔（すなわち、ハードウェア故障かソフトウェア故障かを区別しないシステム故障発生時間間隔）を表す確率変数である。また、この分布関数 $F_{U_n^{sys}}(t) \equiv \Pr\{U_n^{sys} \leq t\}$ は、

$$F_{U_n^{sys}}(t) \equiv \Pr\{U^H \leq t \cup U_n^S \leq t\} \\ = 1 - \Pr\{U^H > t, U_n^S > t\} \\ = 1 - e^{-h_n^{sys} \cdot t} \quad (h_n^{sys} \equiv \lambda_n + \theta; n = 0, 1, 2, \dots), \quad (17)$$

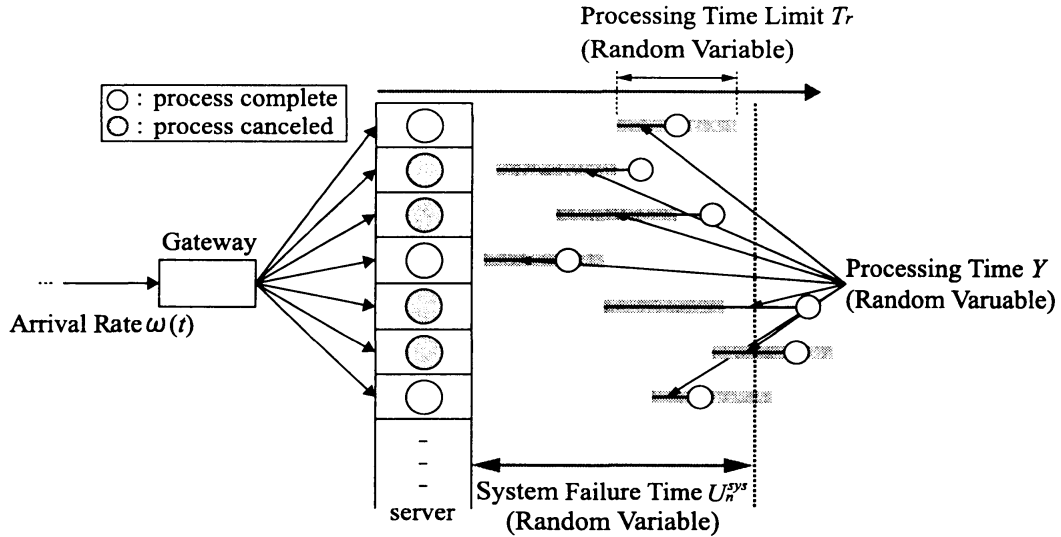


図 2： 確率変数 U_n^{sys}, Y, T_r と仕事処理の完了/破棄の関係

で与えられ、 h_n^{sys} はシステム全体の故障発生率を表す。これ以降、 i 個のフォールトが修正されている状態でシステムが再稼動を始める時点をも、時刻の原点 $t = 0$ として議論を進める。

$\{Z_i^1(t), t \geq 0\}$ を、時刻 t までに到着した仕事のうち、処理時間制約 T_r 以内に完了することができる仕事数を表す確率過程とする。このとき、 $Z_i^1(t)$ の $\{N(t) = k\}$ による条件付き確率を考慮すると、

$$\Pr\{Z_i^1(t) = j\} = \sum_{k=0}^{\infty} \Pr\{Z_i^1(t) = j | N(t) = k\} e^{-\Omega(t)} \frac{[\Omega(t)]^k}{k!}, \quad (18)$$

を得る。システムの状態が $\{X(t) = W_n\}$ のとき、1 つの仕事がシステムに到着し、かつその仕事の処理が処理時間制約までに完了することができる確率は、

$$\begin{aligned} \beta_n &\equiv \Pr\{Y < U_n^{sys}, Y < T_r | X(t) = W_n\} \\ &= \int_0^{\infty} e^{-h_n^{sys} \cdot y} \overline{F_{T_r}}(y) dF_Y(y), \end{aligned} \quad (19)$$

で表される。ここで、 $\overline{F_{T_r}}(y) = 1 - F_{T_r}(y)$ である。また、仮定 B2 より、 $\{N(t) = k\}$ が与えられているとき、任意の 1 つの仕事に対する到着時間は、NHPP の性質より以下の密度関数

$$f(x) = \begin{cases} \frac{\omega(x)}{\Omega(t)} & (0 \leq x \leq t) \\ 0 & (x > t) \end{cases}, \quad (20)$$

をもつ確率分布に従う。よって、時刻 t までに到着した任意の 1 つの仕事が、処理時間制約以内に処理を完了できる確率は、無限サーバ待ち行列理論を適用して、

$$\begin{aligned} p_i^1(t) &= \int_0^t \sum_{n=i}^{\infty} \Pr\{X(x) = W_n | X(0) = W_i\} \cdot \beta_n \cdot f(x) dx \\ &= \frac{1}{\Omega(t)} \sum_{n=i}^{\infty} \beta_n \int_0^t P_{W_i, W_n}(x) \omega(x) dx, \end{aligned} \quad (21)$$

で与えられる。仮定 B3 より、各仕事の処理時間は独立であるので、 $Z_i^1(t)$ の条件付き確率は、

$$\Pr\{Z_i^1(t) = j | N(t) = k\} = \begin{cases} \binom{k}{j} [p_i^1(t)]^j [1 - p_i^1(t)]^{k-j} & (j = 0, 1, 2, \dots, k) \\ 0 & (j > k) \end{cases}, \quad (22)$$

の二項過程となる。式 (22) を式 (18) に代入すると、 $\{Z_i^1(t), t \geq 0\}$ の分布は、

$$\begin{aligned} \Pr\{Z_i^1(t) = j\} &= \sum_{k=j}^{\infty} \binom{k}{j} [p_i^1(t)]^j [1 - p_i^1(t)]^{k-j} e^{-\Omega(t)} \frac{[\Omega(t)]^k}{k!} \\ &= e^{-\Omega(t)p_i^1(t)} \frac{[\Omega(t)p_i^1(t)]^j}{j!}, \end{aligned} \quad (23)$$

となる。式 (23) は、 $\{Z_i^1(t), t \geq 0\}$ が平均値関数 $\Omega(t)p_i^1(t)$ をもつ NHPP に従うことを意味する。

$\{Z_i^2(t), t \geq 0\}$ を、時刻 t までに到着した仕事のうち、処理を完了することができない仕事数を表す確率過程とすると、 $\{Z_i^1(t), t \geq 0\}$ の場合と同様な議論が適用できる。すなわち、 $Z_i^2(t)$ の分布は、

$$\Pr\{Z_i^2(t) = j\} = e^{-\Omega(t)p_i^2(t)} \frac{[\Omega(t)p_i^2(t)]^j}{j!} \quad (j = 0, 1, 2, \dots), \quad (24)$$

$$p_i^2(t) = 1 - p_i^1(t), \quad (25)$$

と得られる。式 (24) は、 $\{Z_i^2(t), t \geq 0\}$ が平均値関数 $\Omega(t)p_i^2(t)$ をもつ NHPP に従うことを意味する。

4 システムの性能評価尺度

上記の解析に基づき、リアルタイム性を考慮したいいくつかの性能評価尺度を得ることができる。時間区間 $(0, t]$ の間に処理を完了することができる仕事数の平均値を表す期待処理可能仕事数は、

$$\Lambda_i^1(t) \equiv E[Z_i^1(t)] = \sum_{n=i}^{\infty} \beta_n \int_0^t P_{W_i, W_n}(x) \omega(x) dx, \quad (26)$$

で与えられる。また、時刻 t における瞬間仕事処理完了率は、

$$\gamma_i^1(t) \equiv \frac{d\Lambda_i^1(t)}{dt} \bigg/ \omega(x) = \sum_{n=i}^{\infty} \beta_n P_{W_i, W_n}(t), \quad (27)$$

となる。式 (27) は、時刻 t において単位時間当りに到着する仕事数に対する処理可能な仕事数の割合を表す。さらに、式 (21) の $p_i^1(t)$ については、

$$p_i^1(t) = \frac{E[Z_i^1(t)]}{E[N(t)]}, \quad (28)$$

と解釈することができる。すなわち、 $p_i^1(t)$ は時間区間 $(0, t]$ の間に到着する仕事数に対する処理可能な仕事数の割合である累積仕事処理完了率を表すことになる。

処理を完了することができない仕事数に関する評価尺度についても、同様に導出することができる。期待損失仕事数、瞬間仕事損失率および累積仕事損失率は、それぞれ、

$$\Lambda_i^2(t) \equiv E[Z_i^2(t)] = \Omega(t) - \sum_{n=i}^{\infty} \beta_n \int_0^t P_{W_i, W_n}(x) \omega(x) dx, \quad (29)$$

$$\gamma_i^2(t) \equiv \frac{d\Lambda_i^2(t)}{dt} \bigg/ \omega(x) = 1 - \sum_{n=i}^{\infty} \beta_n P_{W_i, W_n}(t), \quad (30)$$

$$p_i^2(t) = \frac{E[Z_i^2(t)]}{E[N(t)]}, \quad (31)$$

と得られる。

ただし、式 (26)–(31) をそのまま性能評価尺度として使用することは困難である。なぜならば、上式に現れる初期条件 i は時刻 $t = 0$ におけるフォールト修正数を表しており、不完全デバッグ環境を仮定している本モデルにおいては、この値を直ちに観測することは困難であるからである。しかしながら、ソフトウェア故障発生数あるいはそれに伴うデバッグ作業回数は容易に観測が可能である。また、 l 回目のデバッグ作業が実施された直後に修正されているフォールト数の累積値を表す確率変数 C_l は、次の二項分布に従う。

$$\Pr\{C_l = i\} = \Pr\{X(0) = W_i\} = \binom{l}{i} a^i b^{l-i} \quad (i = 0, 1, 2, \dots, l). \quad (32)$$

したがって、式 (32) を用いて、式 (26)–(31) をデバッグ作業回数 l の関数として表すと、それぞれ

$$\Lambda^1(t, l) = \sum_{i=0}^l \binom{l}{i} a^i b^{l-i} \Lambda_i^1(t) \quad (t \geq 0; \quad l = 0, 1, 2, \dots), \quad (33)$$

$$\gamma^1(t, l) = \sum_{i=0}^l \binom{l}{i} a^i b^{l-i} \gamma_i^1(t) \quad (t \geq 0; \quad l = 0, 1, 2, \dots), \quad (34)$$

$$p^1(t, l) = \frac{1}{\Omega(t)} \sum_{i=0}^l \binom{l}{i} a^i b^{l-i} \Lambda_i^1(t) \quad (t \geq 0; \quad l = 0, 1, 2, \dots), \quad (35)$$

$$\Lambda^2(t, l) = \Omega(t) - \sum_{i=0}^l \binom{l}{i} a^i b^{l-i} \Lambda_i^1(t) \quad (t \geq 0; \quad l = 0, 1, 2, \dots), \quad (36)$$

$$\gamma^2(t, l) = 1 - \sum_{i=0}^l \binom{l}{i} a^i b^{l-i} \gamma_i^1(t) \quad (t \geq 0; \quad l = 0, 1, 2, \dots), \quad (37)$$

$$p^2(t, l) = 1 - \sum_{i=0}^l \binom{l}{i} a^i b^{l-i} p_i^1(t) \quad (t \geq 0; \quad l = 0, 1, 2, \dots), \quad (38)$$

で与えられる．式(33)–(38)は，それぞれ時刻 $t = 0$ において l 回目のデバッグ作業が完了しているとき，その後の時刻 t における期待処理可能仕事数，瞬間仕事完了率，累積仕事処理完了率，期待損失仕事数，瞬間仕事損失率および累積仕事損失率を表す．

5 数値例

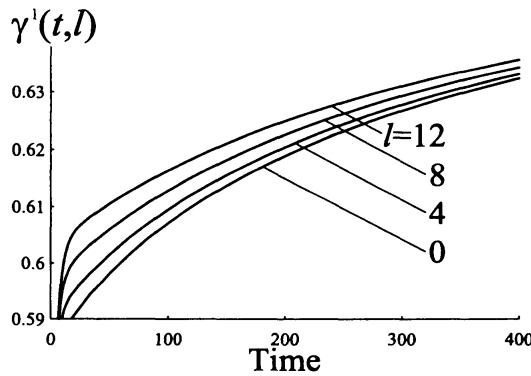


図 3： デバッグ回数 l と $\gamma^1(t, l)$ の関係
($\nu_Y = \nu_{Tr} = 1.0$, $\alpha_Y = 500.0$, $\alpha_{Tr} = 200.0$,
 $a = 0.8$, $\theta = 0.01$, $\eta = 2.0$).

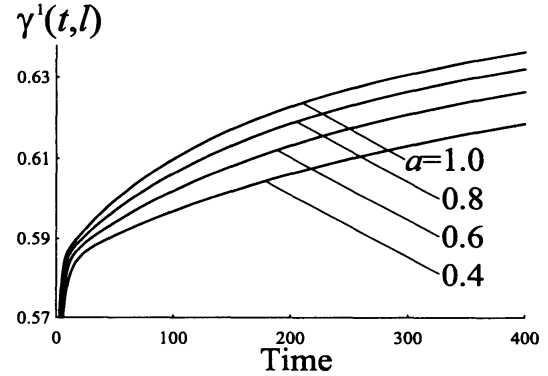


図 4： 完全デバッグ率 a と $\gamma^1(t, l)$ の関係
($l = 0$, $\nu_Y = \nu_{Tr} = 1.0$, $\alpha_Y = 500.0$,
 $\alpha_{Tr} = 200.0$, $a = 0.8$, $\theta = 0.01$, $\eta = 2.0$).

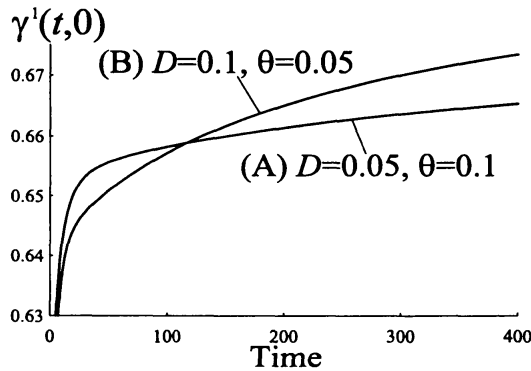


図 5： θ および D と $\gamma^1(t, l)$ の関係 ($l = 0$ のとき)
($\nu_Y = \nu_{Tr} = 1.0$, $\alpha_Y = 500.0$, $\alpha_{Tr} = 200.0$,
 $a = 0.8$, $\eta = 2.0$).

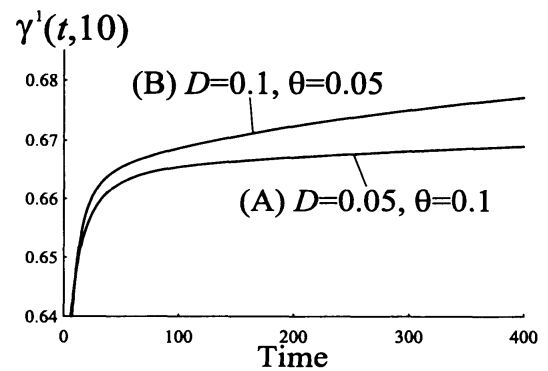


図 6： θ および D と $\gamma^1(t, l)$ の関係 ($l = 10$ のとき)
($\nu_Y = \nu_{Tr} = 1.0$, $\alpha_Y = 500.0$, $\alpha_{Tr} = 200.0$,
 $a = 0.8$, $\eta = 2.0$).

上記の議論に基づくシステムの性能評価例を示す．ここで， λ_n および μ_n には，それぞれ $\lambda_n \equiv Dc^n$ ($D > 0, 0 < c < 1$) および $\mu_n \equiv Er^n$ ($E > 0, 0 < r \leq 1$) が適用される [19]．また， λ_n および μ_n のパラメータ値には，文献 [16] に示される以下の推定値を用いる．

$$\hat{D} = 0.246, \quad \hat{c} = 0.940, \quad \hat{E} = 1.114, \quad \hat{r} = 0.960.$$

また， $F_Y(t)$ および $F_{Tr}(t)$ には，密度関数が以下のように表されるガンマ分布を適用する．

$$\frac{dF_I(t)}{dt} \equiv \frac{dH(t|\nu_I, \alpha_I)}{dt} = \frac{\alpha_I^{\nu_I} t^{\nu_I-1} e^{-\alpha_I t}}{\int_0^\infty x^{\nu_I-1} e^{-x} dx} \quad (t \geq 0; \quad \nu_I > 0, \quad \alpha_I > 0; \quad I \in \{Y, Tr\}). \quad (39)$$

ここで， ν_I および α_I は，それぞれ形状パラメータおよび尺度パラメータである．式(39)の平均および分散は，それぞれ ν_I/α_I および ν_I/α_I^2 で表される．

図3および図4は、それぞれデバッグ回数 l および完全デバッグ率 a と式(34)の瞬間仕事処理完了率 $\gamma(t, l)$ の関係を示す。これらの図より、デバッグ作業が進行するほど、またシステム保守要員のデバッグ能力が高いほど、システム全体の処理性能も向上する様子がわかる。

図5($l=0$ とき)および図6($l=10$ のとき)は、それぞれ次の2つのパラメータの設定値について示している。

case (A): $D = 0.05, \theta = 0.1$, case (B): $D = 0.1, \theta = 0.05$.

$n=0$ のとき、システム全体の故障率は、case (A), case(B)とも同じ $h_0^{ys} = 0.15$ である。しかしながら、システムの性能評価は異なる結果を示している。すなわち、図5では、case (B)の方が、システム稼働の初期段階では性能評価は低いものの、時間の経過とともにcase (A)より高い性能を発揮している様子が示される。また、図6では、示される全時間区間において、case (B)の方が高い性能を発揮している様子が示される。この評価結果の違いは、システム全体の固有信頼度の成長する余裕の多寡による。例えば、 $n=10$ のとき、case (A), case (B)におけるシステム故障率は、それぞれ $h_{10}^{ys} = 0.127$ および $h_{10}^{ys} = 0.103$ である。case (B)の方が運用開始後も信頼度の成長する余地があることが、図5および図6に示される評価結果の違いに表れている。

6 むすび

本論文では、コデザインの概念を導入して、ハードウェアおよびソフトウェアシステムの故障/修復特性を同時に考慮したコンピュータシステムの性能評価法について議論した。処理可能な仕事数に関連するシステムの性能評価尺度を、時間およびソフトウェアシステムに対するデバッグ作業回数の関数として導出した。本モデルを適用することにより、システムの保守要員の能力およびシステム固有の故障/修復特性と、システム全体の性能評価の関係を把握することができる。

謝 辞

本研究の一部は、日本学術振興会科学研究費補助金基盤研究(C)(課題番号 20510136)および高橋産業経済研究財団の援助を受けたことを付記する。

参考文献

- [1] G. De Micheli and R. K. Gupta, "Hardware/software co-design," *Proceedings of the IEEE*, vol. 85, no. 3, pp. 349-365, 1997.
- [2] I. Bolsens, H. J. De Man, B. Lin, K. Van Rompaey, S. Vercauteren and D. Verkest, "Hardware/software co-design of digital telecommunication system," *Proceedings of the IEEE*, vol. 85, no. 3, pp. 391-418, 1997.
- [3] A. L. Goel and J. Soenjoto, "Models for hardware-software system operational-performance evaluation," *IEEE Trans. Reliability*, vol. R-30, no. 3, pp. 232-239, 1981.
- [4] K. Tokuno and S. Yamada, "Markovian availability modeling for software-intensive systems," *International Journal of Quality and Reliability Management*, vol. 17, no. 2, pp. 200-212, 2000.
- [5] U. Sumita and Y. Masuda, "Analysis of software availability/reliability under the influence of hardware failures," *IEEE Trans. Software Engineering*, vol. SE-12, no. 1, pp. 32-41, 1986.
- [6] U. Sumita and M. Kijima, "Theory and algorithms of the Laguerre transform, part 1: theory," *Journal of the Operations Research Society of Japan*, vol. 31, no. 4, pp. 467-495, 1988.
- [7] U. Sumita and M. Kijima, "Theory and algorithms of the Laguerre transform, part 2: algorithm," *Journal of the Operations Research Society of Japan*, vol. 34, no. 4, pp. 449-477, 1991.
- [8] S. Osaki, H. Ohshimo and S. Fukumoto, "Effect of software maintenance policies for a hardware-software system," *International Journal of Systems Sciences*, vol. 20, no. 2, pp. 331-338, 1989.
- [9] S. R. Welke, B. W. Johnson and J. H. Aylor, "Reliability modeling of hardware/software system," *IEEE Trans. Reliability*, vol. 44, no. 3, pp. 413-418, 1995.
- [10] A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," *IEEE Trans. Reliability*, vol. R-28, no. 3, pp. 206-211, 1979.
- [11] C. D. Lai, M. Xie, K. L. Poh, Y. S. Dai and P. Yang, "A model for availability analysis of distributed software/hardware system," *Information and Software Technology*, vol. 44, no. 3, pp. 343-350, 2002.
- [12] G. Levitin, "Reliability and performance analysis of hardware-software system with fault-tolerant software components," *Reliability Engineering & System Safety*, vol. 91, pp. 570-579, 2006.
- [13] A. Avizienis, "The N -version approach to fault-tolerant," *IEEE Trans. Software Engineering*, vol. SE-11, no. 12, pp. 1491-1501, 1985.
- [14] B. Randell, "System structure for software fault-tolerance," *IEEE Trans. Software Engineering*, vol. SE-1, no. 2, pp. 220-232, 1975.
- [15] J. K. Muppala, S. P. Woollet and K. S. Trivedi, "Real-time-systems performance in the presence of failures," *Computer*, vol. 24, no. 5, pp. 37-47, 1991.

- [16] K. Tokuno and S. Yamada, "Stochastic performance evaluation for multi-task processing system with software availability model," *Journal of Quality in Maintenance Engineering*, vol. 12, no. 4, pp. 412–424, 2006.
- [17] S. M. Ross, *Introduction to Probability Models (9th Edition)*, San Diego, Academic Press, 2007.
- [18] S. Osaki, *Applied Stochastic System Modeling*, Heidelberg, Springer-Verlag, 1992.
- [19] P. B. Moranda, "Event-altered rate models for general reliability analysis," *IEEE Trans. Reliability*, vol. R-28, no. 5, pp. 376–381, 1979.

付録 A $G_{i,n}(t)$ の導出

$G_{i,n}(t)$ の L-S 変換は次のように定義できる.

$$\tilde{G}_{i,n}(s) \equiv \int_0^\infty e^{-st} dG_{i,n}(t). \quad (40)$$

式 (7) に L-S 変換を施すことにより, 以下の式を得る.

$$\begin{aligned} \tilde{G}_{i,n}(s) &= \tilde{Q}_{W_i, R_i^S}(s) \tilde{Q}_{R_i^S, W_{i+1}}(s) \tilde{G}_{i+1,n}(s) + \tilde{Q}_{W_i, R_i^S}(s) \tilde{Q}_{R_i^S, W_i}(s) \tilde{G}_{i,n}(s) \\ &\quad + \tilde{Q}_{W_i, R_i^H}(s) \tilde{Q}_{R_i^H, W_i}(s) \tilde{G}_{i,n}(s) \quad (i = 0, 1, 2, \dots, n-1). \end{aligned} \quad (41)$$

式 (2)–(6) の L-S 変換は, それぞれ以下のように与えられる.

$$\tilde{Q}_{W_i, R_i^S}(s) = \frac{\lambda_i}{s + \theta + \lambda_i}, \quad (42)$$

$$\tilde{Q}_{W_i, R_i^H}(s) = \frac{\theta}{s + \theta + \lambda_i}, \quad (43)$$

$$\tilde{Q}_{R_i^S, W_{i+1}}(s) = \frac{a\mu_i}{s + \mu_i}, \quad (44)$$

$$\tilde{Q}_{R_i^S, W_i}(s) = \frac{b\mu_i}{s + \mu_i}, \quad (45)$$

$$\tilde{Q}_{R_i^H, W_i}(s) = \frac{\eta}{s + \eta}. \quad (46)$$

式 (42)–(46) を式 (41) に代入し整理すると,

$$\tilde{G}_{i,n}(s) = \frac{a\lambda_i\mu_i(s+\eta)}{(s+d_i^1)(s+d_i^2)(s+d_i^3)} \tilde{G}_{i+1,n}(s) \quad (i = 0, 1, 2, \dots, n-1), \quad (47)$$

となる. 式 (47) を再帰的に解くと, $G_{i,n}(t)$ の L-S 変換は,

$$\tilde{G}_{i,n}(s) = \prod_{k=i}^{n-1} \frac{a\lambda_k\mu_k(s+\eta)}{(s+d_k^1)(s+d_k^2)(s+d_k^3)} = \sum_{k=i}^{n-1} \left(\frac{A_{n,i}^1(k)d_k^1}{s+d_k^1} + \frac{A_{n,i}^2(k)d_k^2}{s+d_k^2} + \frac{A_{n,i}^3(k)d_k^3}{s+d_k^3} \right), \quad (48)$$

と得られる. 式 (48) を t について逆変換することにより, 式 (8) を得る. このとき, 式 (10)–(12) は, 次式によって得られる.

$$A_{i,n}^j(k) = \frac{1}{d_k^j} (s+d_k^j) \tilde{G}_{i,n}(s) \Big|_{s=-d_k^j} \quad (j = 1, 2, 3; k = i, i+1, \dots, n-1). \quad (49)$$

付録 B $P_{W_i, W_n}(t)$ の導出

式 (15) より, $P_{W_n, W_n}(t)$ の L-S 変換は, 以下のように得られる.

$$\tilde{P}_{W_n, W_n}(s) = \frac{s(s+\mu_n)(s+\eta)}{(s+d_n^1)(s+d_n^2)(s+d_n^3)}. \quad (50)$$

式 (14) を L-S 変換し, これに式 (50) を代入すると,

$$\tilde{P}_{W_i, W_n}(s) = \frac{s(s+\mu_n)}{a\lambda_n\mu_n} \tilde{G}_{i,n+1}(s) = \left(\frac{s}{a\lambda_n} + \frac{s^2}{a\lambda_n\mu_n} \right) \tilde{G}_{i,n+1}(s), \quad (51)$$

となる. 式 (51) を t について逆変換することにより, 式 (16) を得る.